

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-56886

(43) 公開日 平成7年(1995)3月3日

(51) Int.Cl. ⁹	識別記号	片内整理番号	F I	技術表示箇所
G 0 6 F 15/82		9289-5L		
15/16	3 9 0 Z	7429-5L		

審査請求 未請求 請求項の数1 O L (全 11 頁)

(21) 出願番号 特願平5-206292

(22) 出願日 平成5年(1993)8月20日

(71) 出願人 000005234

富士電機株式会社

神奈川県川崎市川崎区田辺新田1番1号

(71) 出願人 000237156

富士ファコム制御株式会社

東京都日野市富士町1番地

(72) 発明者 河原 規明

東京都日野市富士町1番地 富士ファコム
制御株式会社内

(72) 発明者 安西 文彦

東京都日野市富士町1番地 富士ファコム
制御株式会社内

(74) 代理人 弁理士 大曾 義之

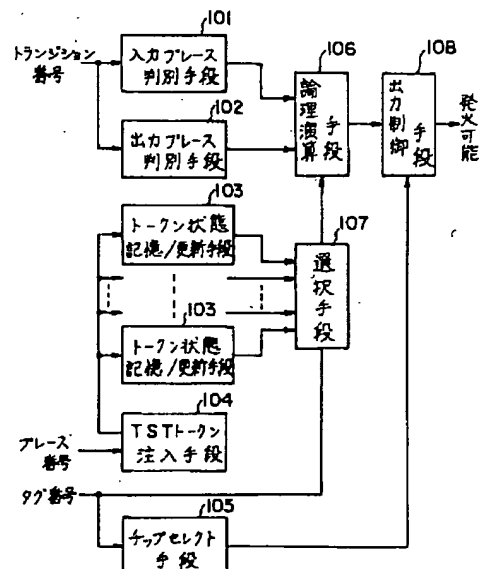
(54) 【発明の名称】 ペトリネットに基づく並列制御システム

(57) 【要約】 (修正有)

【目的】 入れ子構造が多い場合も、発火チェックが行える。

【構成】 プロセッサの実行状態を監視する状態チェック装置502を階層化された順序記述で制御する実行順序制御装置503における発火回路において、トランジションに接続する入力ブレースの情報を格納する入力ブレース判別手段101、トランジションに接続する出力ブレースの情報を格納する出力ブレース判別手段102、ブレースのトークンの状態を階層毎に記憶/更新するトークン状態記憶/更新手段103、ブレースのトークンの注入を、各階層のトークン状態記憶/更新手段に対して行うことが可能なTSTトークン注入手段104、複数のトークン状態記憶/更新手段の中から1つを選択する選択手段107、実行順序制御装置の出力を有効/無効にする出力制御手段108、及び発火可能か否かを判別する論理演算手段106を備える。

本発明のブロック図



【特許請求の範囲】

【請求項1】 各々が、独自に異なる処理をすることが可能で、内部又は外部にローカルにアクセスすることができるメモリを持つ複数のプロセッサ(501)と、前記プロセッサ(501)の実行状態を監視する状態チェック装置(502)と、

前記プロセッサ(501)及び前記状態チェック装置(502)を階層化された順序記述で制御する実行順序制御装置(503)と、

を有するマルチプロセッサの並列制御システムにおいて、

前記実行順序制御装置(503)内に、

トランジションに接続する入力ブレースの情報を記憶する入力ブレース判別手段(101)と、

トランジションに接続する出力ブレースの情報を記憶する出力ブレース判別手段(102)と、

ブレースのトークンの状態を、階層毎に記憶/更新する複数のトークン状態記憶/更新手段(103)と、

ブレースのトークンの注入を、各階層の前記トークン状態記憶/更新手段に対して行うことが可能なTSTトークン注入手段(104)と、

前記複数のトークン状態記憶/更新手段の中から1つを選択する選択手段(107)と、

前記実行順序制御装置(503)をセレクトする、チップセレクト手段(105)と、

*

ベトリネットとは、次の4項組Nのことをいう。

$$N = \{ P, T, A, M \}$$

ここで、

ブレースの有限集合

$$P = \{ p_1, p_2, \dots, p_m \}$$

トランジションの有限集合

$$T = \{ t_1, t_2, \dots, t_n \}$$

有向枝（アーク）の有限集合

$$A \subseteq \{ P \times T \} \cup \{ T \times P \}$$

初期マーキング

$$M : P \rightarrow \{ 0, 1, 2, \dots \}$$

初期マーキングMは各ブレースに対してトークンの数を対応させる写像であり

、具体的にはベクトル $(M(p_1), M(p_2), \dots, M(p_m))$ で表される。

【0005】図6(a)は、ベトリネットグラフの一例である。図6(a)を例にとって前述の定義を当てはめると、以下ようになる。

【0006】

【数2】

* 前記実行順序制御装置(503)の出力を、有効/無効にする出力制御手段(108)と、

発火可能か否かを判別する論理演算手段(106)と、を具備することを特徴とするベトリネットに基づく並列制御システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、複数のプログラムモジュール（タスク、サブルーチン、ステートメント、関数などのデータ処理の単位）を、複数のプロセッサによって同時に処理をする、並列処理の並列実行順序制御システムに関する。

【0002】

【従来の技術】従来のベトリネットによる並列制御システムを図5及び図6によって説明する。図5は、並列制御システムを示しており、複数のプロセッサPE(501)、状態チェック装置CCM(502)、実行順序制御装置NCE(503)から構成されている。各PEは、それぞれローカルにアクセスできるメモリを持っており、そのメモリに処理すべき複数のプログラムモジュールが記憶されている。

【0003】ベトリネットグラフは、形式的に以下のよう定義される。

【0004】

【数1】

例えばブレース p_1 からトランジション t_1 への有向枝を、 $p_1 \rightarrow t_1$ と表すとし、さらにブレース p_2 及び p_3 にのみトークンが存在するとすれば、

ブレースの有限集合 $P = \{ p_1, p_2, p_3, p_4 \}$
 トランジションの有限集合 $T = \{ t_1, t_2 \}$
 有向枝 (アーク) の有限集合 $A = \{ p_1 \rightarrow t_1, t_1 \rightarrow p_2, t_1 \rightarrow p_3, p_2 \rightarrow t_2, p_3 \rightarrow t_2, t_2 \rightarrow p_4 \}$
 初期マーキング $M = \{ 0, 1, 1, 0 \}$

(但し、ブレースに存在するトークンの数は1以下であり、マーキングは後述する発火によって変化する。)

【0007】各ブレース $p_1 \sim p_4$ には、プロセッサが処理すべきプログラムモジュールが割り当ててあり、ブレースがトークンを得たときに、プロセッサはそのプログラムモジュールの処理を開始する。

【0008】プログラムモジュールの状態遷移を記述したベトリネットグラフは表形式に変換され、FCT (Fire Condition Table)、TTT (Token Transfer Table) としてNCE内のレジスタに格納されている。

【0009】FCTはブレースに接続している出力トランジションの状態を表し、図6(a)のグラフを表現すると図6(b)のような表になる。TTTはトランジションに接続している出力ブレースの状態を表し、図6(a)のグラフを表現すると図6(c)のような表になる。

【0010】又、ベトリネットの状態を表すトークンは、各ブレース毎にトークンの有り無しをTST (Token Status Table) (103) に格納している。TSTはNCE内のレジスタであり、ブレース毎に1ビットが割りつけられていて、トークン有りて '1'、トークン無しで '0' が格納される。

【0011】NCEはFCT、TTT及びTSTを参照して、発火可能なトランジションを発火させる。発火とは、トランジションの入力ブレースの全てにトークンが入っているときに、入力ブレースからトークンを抜き取り、出力ブレースへトークンを分配 (ここでは、出力ブレースのプログラムモジュールに起動をかけること) することを意味する。

【0012】トークンを分配すべきブレースはTTTから求まる。トークンが分配されると、分配されたブレースはプログラムモジュールの処理を開始する。NCEはトランジションの発火により処理すべきブレースの番号を、CCM内のEXQ (Execution Queue) (504) に書き込む。複数のPEは、EXQ内のブレースの番号を競合しながら読み出し、これに対応するプログラムモジュールを処理する。処理を終了したならば、そのブレースの番号をCCM内のEDQ (End Queue) (505) に書き込む。NCEはEDQ内のブレースの番号を取り出し、それに対応するTSTのビットを '1' にする。

【0013】NCEと各PEは、CCMをインタフェースとして独立に動作している。NCEはベトリネットグラフの記述に従い、処理可能なプログラムモジュールに対応するブレースの番号をEXQに書き込む。

【0014】一方、PEはEXQからブレースの番号を読み出して、それに対応するプログラムモジュールを処理するので、PEの数だけプログラムモジュールが並列に動作することが可能である。

【0015】NCEが発火チェックを行うのは、トークンの状態が変化したときでよく、それはつまりTSTにビットが書き込まれたときであり、発火チェックをするトランジションは、トークンが分配されたブレースからアークの伸びているトランジションである。このようにして、ベトリネットに記述された並列処理が実行制御される。

【0016】以下、発火可能かをチェックする発火論理回路について説明する。発火論理回路はNCE内に用意されており、入力ブレース判別手段 (IT)、出力ブレース判別手段 (OT) 及びTSTの3つのレジスタと論理演算部から構成されている。

【0017】ITはトランジション毎の入力ブレースの接続情報を表したレジスタで、FCTを参照して作成可能であり、トランジションに接続している入力ブレースは '1' で表され、接続していないものは '0' で表される。

【0018】図6(a)のベトリネットグラフのITは、図6(d)のように表される。OTはトランジション毎の出力ブレースの接続情報を表したレジスタで、TTTを参照して作成可能であり、トランジションに接続している出力ブレースは '1' で表され、接続していないものは '0' で表される。

【0019】図6(a)のベトリネットグラフのOTは、図6(e)のように表される。発火論理回路はNCE内で、新たにトークンが入ったブレースを入力ブレースとするトランジションの番号を受け取ると、NCE内に保存している情報 (TST、IT、OT) を用いてそのトランジションが発火可能かどうかのチェックを行う。

【0020】全ての入力ブレースにトークンが存在するか否かである。この際に、ネットの安全性のチェックも行う。ネットの安全性とは、ブレースに1つを越えるトークンが入らないことを意味する。

【0021】具体的に言えば、出力ブレースにトークンが入っているうちは、トランジションが発火できなくすることである。そのために、トランジションの入力ブレースのトークンのチェックのみではなく、出力ブレースのトークンのチェックも行う。以上のチェックで発火可能となるとそのトランジションは発火し、入力側の全てのブレースのトークンが消え、出力側の全てのブレースにトークンが入る。

【0022】その結果、TSTが更新される。

発火・更新論理

(入力条件) 全ての p_i に対して $tst(p_i) + \overline{it(t_j, i)} = '1'$
 (出力条件) 全ての p_i に対して $tst(p_i) \times ot(t_j, i) = '0'$
 (TST更新) $tst(p_i) \times \overline{it(t_j, i)} \rightarrow tst(p_i) \quad (i=1, 2, \dots, m)$
 (但し、+は論理和、 \times は論理積を表す。)

【0024】図7にブレース数=4、トランジション数=2のときの発火論理回路を、図8にTST更新回路を示す。図6(a)のベトリネットグラフを使用して以下の説明を行う。

【0025】図7において、トランジション番号がITに入力されると、そのトランジションに接続される入力ブレースの接続情報が出力される。仮にトランジション番号としてt1が入力されると、 $it(t1, 1)$ のみが'1'となりインバータINV4のみが'0'となり、他のインバータは全て'1'となる。又、ブレースp1のみにトークンが存在するとすれば、TST出力の内 $tst(p1)$ のみが'1'となり他の出力は'0'となる。その結果ANDゲートの出力は全て'1'となり、入力条件であるAND4の出力が'1'となり、入力条件は発火可能となる。

【0026】同様に、トランジション番号がOTに入力されると、そのトランジションに接続される出力ブレースの接続情報が出力される。トランジション番号としてt1が入力されると、 $ot(t1, 2)$ と $ot(t1, 3)$ のみが'1'となる。又、トークンはp1のみに存在するのであるから、 $tst(p2)$ と $tst(p3)$ は'0'となりNANDゲートの出力は全て'1'となってAND5の出力が'1'となり、出力条件も発火可能となる。その結果、AND6の出力は'1'となって、トランジションt1は発火可能となる。

【0027】図8は、ITの出力を反転したものと、TSTの出力の論理積をとったものをTSTに格納している。トークンはp1のみに存在するのであるから、TSTの出力の内、 $tst(p1)$ のみが'1'となり、他は全て'0'となる。又、ITの内 $it(t1, 1)$ のみが'1'とな

* ブレースの有限集合 $P = \{ p1, p2, \dots, p_i, \dots, p_m \} \quad (1 \leq i \leq m)$
 トランジションの有限集合 $T = \{ t1, t2, \dots, t_j, \dots, t_n \} \quad (1 \leq j \leq n)$
 TSTレジスタ $TST = (tst(p1), tst(p2), \dots, tst(p_m))$
 トランジション t_j のITレジスタ = $(it(t_j, 1), it(t_j, 2), \dots, it(t_j, m))$
 トランジション t_j のOTレジスタ = $(ot(t_j, 1), ot(t_j, 2), \dots, ot(t_j, m))$
 とすると、トランジション t_j が発火可能であるための条件は、

【0023】

【数3】

り他は全て'0'となる。そして、インバータの内INV5の出力のみが'0'となり、その他のインバータは'1'となる。その結果、ANDゲートの出力は全て'0'となり、p1のトークンが消去されたことになる。

【0028】図9は、OTの出力とTSTの出力の論理和をとり、TSTに格納するものである。OTにトランジション番号1が入力されると、 $ot(t1, 2)$ と $ot(t1, 3)$ のみが'1'となり、他のOTの出力は'0'となる。又、TSTの出力は $tst(p2)$ と $tst(p3)$ が'0'となっており、OR6及びOR7の出力は'1'となり、TSTにはブレース2とブレース3に相当するビットに'1'が格納される。その結果、ブレース2とブレース3にトークンが注入されたことになる。ブレース2とブレース3以外のビットには影響を与えない。

【0029】なお、TSTの状態に変化があるにもかかわらず、トランジションが発火出来ない場合が2種類ある。まず、入力ブレースのいくつかに入力ブレースに入っていないときである。

【0030】この場合、残りの入力ブレースにトークンが分配されてきたときの発火チェックで発火の判定が可能であるので、問題はない。もう1つは入力側ブレースのトークンは全てあるが、出力側ブレースにもトークンがあるときである。これは、出力側ブレースのトークンがなくなるまで待たなければならない。また、なくなり次第、発火できなければならない。しかし、ベトリネット記述をしたテーブルFCFが入力ブレースから発火チェックをするトランジションを引く形なので、出力ブレースの変化からそのブレースの番号をもとに発火チェックをすべきトランジションを引くことは難しい。

【0031】従って、出力ブレースが空くまで入力ブレースの1つを用いてそのトランジションの発火チェックを繰り返せばよい。但し、一旦入れたトークンが他のトランジションの発火に影響を与える場合もあるので、強制的にこのトークンを消去して、そのブレース番号をEDQに戻してやる。こうすると、NCEはそのブレースのプログラムモジュールの処理が終了したものとみなし、TSTのそのブレースのビットを「1」にする。

【0032】その結果、出力ブレースが空くまでそのブレース番号はNCEとEDQを往復して発火チェックを繰り返す。又、ベトリネットグラフは、モジュール性を持たせるために入れ子構成が組めるようになっている。その機能を、図10～図13を用いて以下に述べる。

【0033】図10に、入れ子構成にしたベトリネットグラフを示す。NCE内の記憶部に表形式で格納されたベトリネットグラフ情報を、ネットプログラムと呼ぶ。ネットプログラムにモジュール性を持たせるために、NCEはサブネットコール（サブルーチンコール）を行うことができる。すなわち、1つのサブネット全体を1つのブレースとして扱うことができる。図10(a)の中央及び右側のネットがそれぞれサブネットになっている。ブレースp3及びp7をコールブレースと呼び、これは従来のブレース（以後通常ブレースと呼び、プログラムモジュールが割り付けられているブレースである）とは違い、プログラムモジュールは割り付けられていない。サブネットの開始点、終了点を与えるブレースとして、スタートブレース、エンドブレースと呼ばれるブレースがある。

【0034】図10(a)のブレースp1、p5、p9は、それぞれスタートブレースであり、プログラムモジュールを割り付けてもよいし、割り付けなくてもよい。図10(a)のブレースp4、p8、p12は、それぞれエンドブレースであり、プログラムモジュールは割り付けられない。

【0035】トランジションの発火によって、コールブレースにトークンが入ると、制御はサブネットに移る。そして、サブネットのエンドブレースにトークンを入れるとき、制御は元のネットに戻る。NCEは、FCT、TTTの他にPAT（Placeattribute Table）を持つ。図10(a)のグラフのPATを図10(b)に示す。

【0036】NCEはこのテーブルによって、ブレース番号からそのブレースの属性（通常ブレース、スタートブレース、エンドブレース、コールブレース）を参照することができる。NCEは、トランジションの発火によって次に実行するブレースの番号を知ったとき、PATを用いてそのブレースの属性を知り、その属性に応じた処理を行う。PATで、属性がコールブレースである所には、サブネットのスタートブレース番号と一緒に格納されている。

【0037】コールブレースによって制御をサブネット

に移す際、NCEはネット毎の識別子として、タグと呼ぶ番号を内部生成する。これは、ネットプログラムが並列プログラムであるため、同一のサブネットを呼ぶコールブレースが複数個起動されることがあり得るからである。さらに、NCEはサブネットのネスト構造の親子関係を保持するため、タグ番号をリストの形で記憶する。

【0038】このリストをタグリストと呼び、NCE内に置く。図10(a)のグラフのタグリストを、図10(c)に示す。タグリストには、呼び出し元のタグ番号とコールブレース番号が、同時に保持される。又、トランジションの発火によって、次に実行するブレース番号をEXQに書き込む際、一緒にタグ番号を書き込む。なお、システム全体の初期化後、最初に実行が始まるネットプログラム（メインネット）には、常にタグ番号“0”が与えられる。

【0039】タグを導入するにあたって、発火回路内にあるTSTは、呼び出し元のTSTを保存するために、タグ番号毎に別個のものがようになる。よって、発火回路での発火チェックの際には、タグ番号を用いて必要なTSTを選び使用する。

【0040】又、トランジションが発火したときのTSTの更新は、タグ番号で選ばれたTSTについてのみ行う。さらに、NCEがEDQから終了したブレースプログラムのタグ番号とブレース番号を読みだしたときの、TSTへのトークン注入は、そのタグ番号が選ぶTSTに対して行う。

【0041】図11～図13に、タグ数=4を扱える発火論理回路、TST更新回路、TSTトークン注入回路を示す。以下、図10(a)のグラフを使用して説明する。トランジションt1が発火すると、ブレース2とブレース3にトークンが注入される。NCEはp2が通常ブレースであることを、PATによって知った際、EXQにブレース番号2とタグ番号0（メインネットであることを示す）を書き込む。ブレース3は、PAT参照によりコールブレースであることを知った際（サブネットコール）、NCEは新たにタグ番号を生成（ここでタグ番号1を生成し、呼び出し元のタグ番号は0となる）し、そのタグ番号0をコールブレース番号3と共にタグリストに付加する（図10(c)参照）。

【0042】そして、そのタグ番号1を用いて発火回路内にあるタグ番号1に対応したTSTを選び、初期化した後、PATから知ることができるスタートブレース番号5を、新たに生成したタグ番号1と共にEXQに書き込む。プロセッサがプログラムモジュール5（ブレース5にプログラムモジュールがあった場合）の処理を終了すれば、ブレース5にトークンを注入し、トランジション3が発火可能であるか否かをチェックする。

【0043】この際、図11の発火回路の中で、タグ番号1に対応したTSTがセレクタによって選択され、図7で説明した発火論理と全く同じ処理が行われる。又、

図12のTST更新回路の中で、タグ番号1に対応したTSTがセレクトによって選択され、図8で説明したTSTの更新と全く同様のことが行われる。

【0044】又、図13のTSTトークン注入回路の中で、タグ番号1に対応したTSTがセレクトによって選択され、図9で説明したトークン注入と全く同様のことが行われる。

【0045】トランジション3が発火すると、ブレース6とブレース7の属性をPATを参照して調べ、ブレース7がコールブレースであることを知ると、上述ブレース3の場合と同じ処理を行う。この時、タグ番号2が生成され、タグ番号2に対応したTSTを初期化し、タグリストにはタグ番号1とコールブレース番号7が格納される(図10(c)参照)。

【0046】又、TSTはタグ番号2に対応するTSTが使用される。以下同様の処理を行い、ブレース9～11の処理終了後、トランジション6が発火可能であれば発火する。

【0047】ブレース12がエンドブレースであることをPATによって知ると、タグリストから呼び出し元のタグ番号1、コールブレース番号7が読みだされ、コールブレース7にトークンが注入され、TSTはタグ番号1に対応するTSTに復帰する。そして、トランジション4が発火可能か否かをチェックして、発火が可能であれば、前述と同様の動作を行いメインネットに制御が移る。

【0048】又、メインネットにおいてPATによって次に実行するブレースが、エンドブレースであることを知った際は、全ての制御が停止(終了)する。

【0049】

【発明が解決しようとする課題】従来装置において、発火回路をハードウェア化して機能メモリにする場合、1チップに入れられるゲート数には制限があるため、入れ子(タグの数)には限りがあった。それは、入れ子構造の多いネットプログラムを制御するには、入れ子毎に別個のTSTが必要となり、必要なレジスタ数が増加し、その分多くのゲート数が必要となるからである。即ち、従来装置でタグ数に制限があり、入れ子構造の多いネットプログラムが制御できない、という問題があった。

【0050】本発明の課題は、タグ数の拡張を可能にし、カスケード接続で必要なタグ数を実現できる発火チェック回路を提供することにある。

【0051】

【課題を解決するための手段】図1は、本発明のブロック図である。入力ブレース判別手段(101)は、トランジションに接続する入力ブレースの状態であるITの内容が記憶され、メモリ、ラッチ回路、F/F回路等で構成される。

【0052】出力ブレース判別手段(102)は、トランジションに接続する出力ブレースの状態であるOTの内容

が記憶され、メモリ、ラッチ回路、F/F回路等で構成される。

【0053】トークン状態記憶/更新手段(103)は、それぞれのブレースのトークン状態が記憶され、発火が行われると、その発火に関連するブレースのトークン状態が更新される。又、トークン状態記憶/更新手段は、階層構造の各階層ごとにそれぞれ1つずつ割り付けられ、論理ゲートとメモリ、ラッチ回路、F/F回路等で構成される。

【0054】TSTトークン注入手段(104)は、各トークン状態記憶/更新手段にトークンを注入するためのもので、論理ゲート等で構成される。選択手段(107)は、複数のトークン状態記憶/更新手段の中から、1つを選択するためのもので、セレクト等で構成される。

【0055】チップセレクト手段(105)は、実行順序制御装置をセレクトするためのもので、複数個の実行順序制御手段の中から1つをセレクトし、そのチップを有効にするためのものであり、論理ゲート等で構成される。

【0056】論理演算手段(106)は、指定されたトランジションが発火可能か否かを判別するためのものであり、論理ゲート等で構成される。出力制御手段(108)は、実行順序制御装置からの出力を、有効/無効にするためのもので、論理ゲート等で構成される。

【0057】

【作用】トランジションの発火チェックを行う際に、タグ番号の下位ビットによって、チップ内のタグ番号に対応するトークン状態記憶/更新手段を、選択手段によって切り替えて、論理演算回路に入力してチェックを行う。この時、タグ番号の上位ビットを外部のデコーダに入力することによって、どのチップをセレクトするかを決める。セレクトされたチップは、論理演算回路からの出力を、出力制御手段によって有効にし、他のチップの論理演算回路からの出力を無効にする。そして外部で、それぞれの論理演算結果の論理積をとることによって、発火可能か否かが決まる。

【0058】同様に、TSTのトークンを更新する際に、外部で生成されたチップセレクト信号によって、1つのチップがセレクトされ、タグ番号の下位ビットによって、チップ内のタグ番号に対応するトークン状態記憶/更新手段を、選択手段によって切り替えて、そのトークンの状態を更新する。セレクトされなかったチップに対しては、どのトークン状態記憶/更新手段も更新されない。

【0059】同様に、TSTにトークンを注入する際に、外部で生成されたチップセレクト信号によって、1つのチップがセレクトされ、タグ番号の下位ビットによって、チップ内のタグ番号に対応するトークン状態記憶/更新手段を、選択手段によって切り替えて、トークンを注入する。セレクトされなかったチップに対しては、どのトークン状態記憶/更新手段もトークンの状態は変

わらない。

【0060】

【実施例】図2に、2つのチップのカスケード接続によって、タグ数=8に拡張した時の発火論理回路を示す。

【0061】タグ番号の上位ビットを入力した外部のデコーダ(201)によって、仮にチップ1がセレクトされるとすると、インバータINV1の入力のみが“1”となり、チップ2のインバータの入力は“0”となる。

【0062】タグ番号の下位ビットは、チップの内部に 10 入力され、どのタグ番号に対応するTST(103)がセレクトされたかを、セレクト(107)によって切り替える。そして、発火論理にIT(101)、OT(102)と共に入力して、図7で説明したのと全く同じ方法によって発火チェックを行う。発火論理の出力は、OR1に入力される。もう一方の入力は、INV1の出力“0”が入力されるので、発火論理の出力はそのまま外部に出力される。又、チップ2の出力はORによって、デフォルト値“1”が出力され、外部のAND1はチップ1からの出力がそのまま出力されることになる。

【0063】図3に、2つのチップのカスケード接続に 20 によって、タグ数=8に拡張した時のTST更新回路を示す。タグ番号の上位ビットを入力した外部のデコーダ(201)によって、仮にチップ1がセレクトされるとすると、インバータINV2の入力のみが“1”となり、チップ2のインバータの入力は“0”となる。

【0064】タグ番号の下位ビットは、チップの内部に 30 入力され、どのタグ番号に対応するTST(103)がセレクトされたかを、セレクト(107)によって切り替える。そして、INV2の出力は“0”であるので、OR2の出力はINV3の出力がそのまま出力される。従って、デコーダ(301)でセレクトされたTSTに対して、図8で説明したのと全く同じ方法でTSTの更新が行われる。

【0065】又、チップ2のORの出力は“1”であるので、セレクトの出力がそのままTSTに入力されるため、TSTの更新は行われない。図4に、2つのチップのカスケード接続によって、タグ数=8に拡張した時のTSTトークン注入回路を示す。

【0066】タグ番号の上位ビットを入力した外部のデ 40 コーダ(201)によって、仮にチップ1がセレクトされるとすると、チップ1側の出力のみが“1”となり、チップ2側の出力は“0”となる。

【0067】タグ番号の下位ビットは、チップの内部に 入力され、どのタグ番号に対応するTST(103)がセレクトされたかを、セレクト(107)によって切り替える。そして、AND3のデコーダ側の入力は“1”であるので、AND3の出力はOTの出力がそのまま出力される。従って、デコーダ(301)でセレクトされたTSTに

対して、図9で説明したのと全く同じ方法でTSTにトークンが注入される。又、チップ2のANDの出力は“0”であるので、セレクトの出力がそのままTSTに 入力されるため、TSTへのトークンの注入は行われない。

【0068】

【発明の効果】本発明を用いれば、タグ数の拡張が可能となり、カスケード接続により必要なタグ数を持った発火論理回路、TST更新回路、TSTトークン注入回路を提供することが可能になる。そして、ネスト構造の多いモデルのグラフ表現に適用することができるように った。

【図面の簡単な説明】

【図1】本発明の並列制御方式の論理ブロック図である。

【図2】本発明の発火論理回路である。

【図3】本発明のTST更新回路である。

【図4】本発明のTSTトークン注入回路である。

【図5】並列制御システムの構成図である。

【図6】(a)はベトリネットグラフの一例であり、(b)はベトリネットグラフのFCTであり、(c)はベトリネットグラフのTTTであり、(d)はベトリネットグラフのITであり、(e)はベトリネットグラフのOTであ 50 る。

【図7】従来の発火論理回路である。

【図8】従来のTST更新回路である。

【図9】従来のTSTトークン注入回路である。

【図10】ネスト構造のベトリネットグラフの一例である。

【図11】従来のタグを導入した発火論理回路である。

【図12】従来のタグを導入したTST更新回路である。

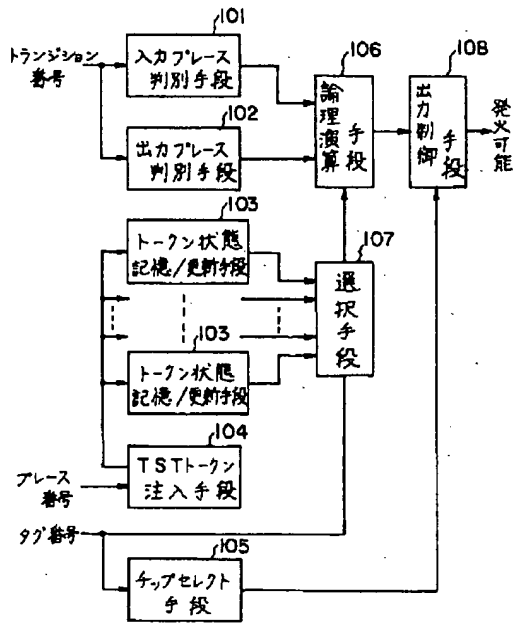
【図13】従来のタグを導入したTSTトークン注入回路である。

【符号の説明】

- 101 入力ブレース判別手段
- 102 出力ブレース判別手段
- 103 トークン状態記憶/更新手段
- 104 TSTトークン注入手段
- 105 チップセレクト手段
- 106 論理演算手段
- 107 選択手段
- 108 出力制御手段
- 501 プロセッサ
- 502 状態チェック装置
- 503 実行順序制御装置
- 504 Execution Queue
- 505 End Queue

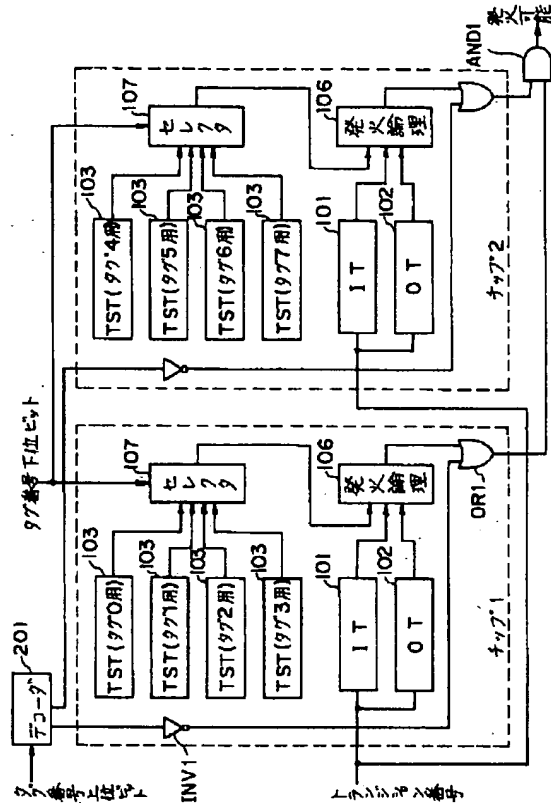
【図1】

本発明のブロック図

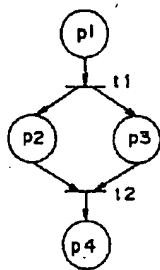


【図2】

本発明の発火論理回路



【図6】



(a) Petriネットグラフの一例

トランジション	トランジション番号
t1	1
t2	2

(b) Petriネットグラフ70PCT

トランジション	トランジション番号
t1	1
t2	2

(c) Petriネットグラフ70TTT

トランジション	p1	p2	p3	p4
t1	1	0	0	0
t2	0	1	1	0

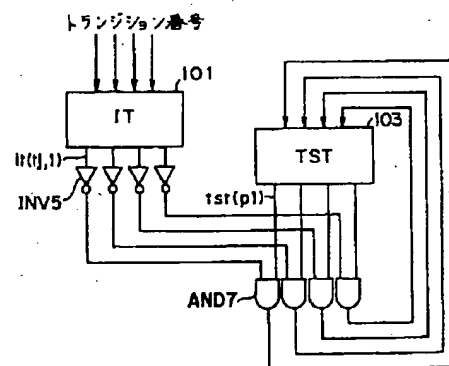
(d) Petriネットグラフ70IT

トランジション	p1	p2	p3	p4
t1	1	0	0	0
t2	0	1	1	0

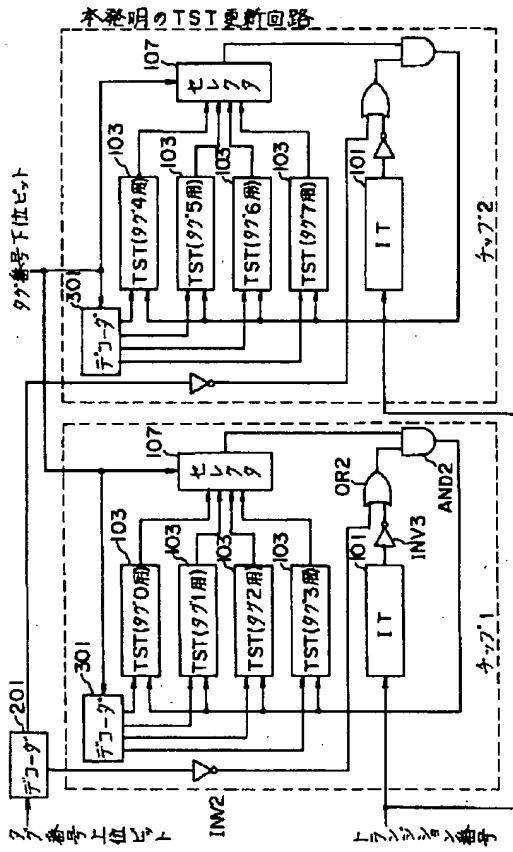
(e) Petriネットグラフ70OT

【図8】

従来のTST更新回路

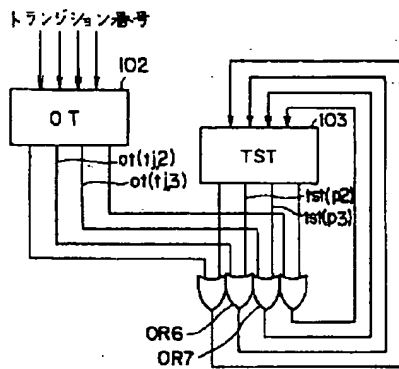


【図3】

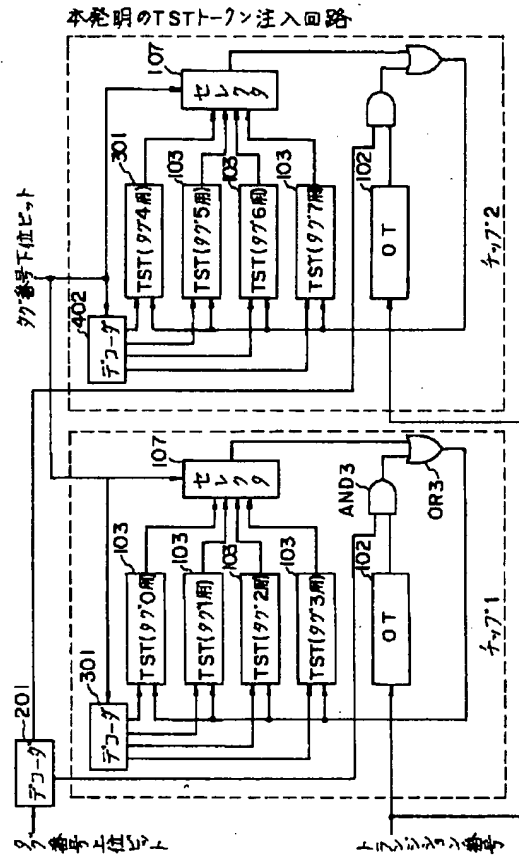


【図9】

従来のTSTトークン注入回路

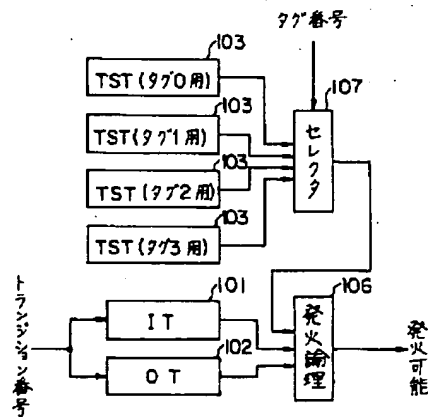


【図4】



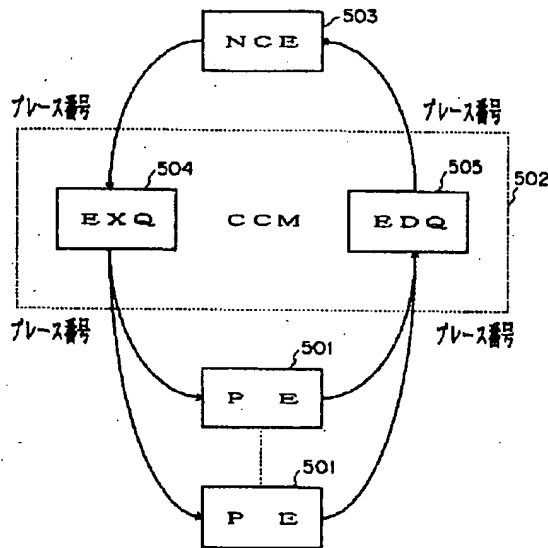
【図11】

従来のタグを導入した発火論理回路



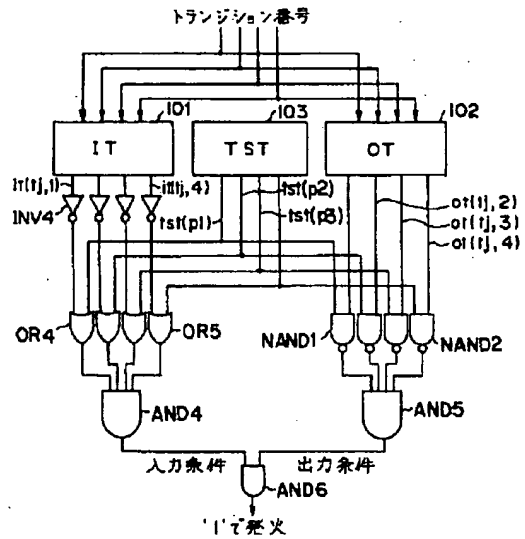
【図5】

並列制御システム

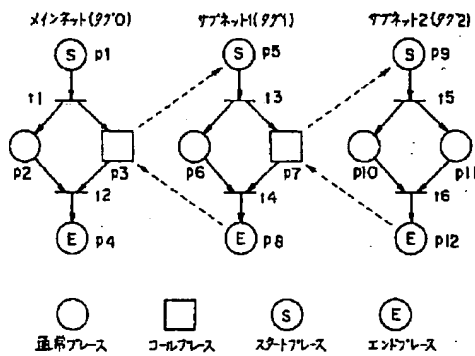


【図7】

従来の発火論理回路



【図10】



(a) ネット構造のペトリネットグラフ例

プレス番号	属性
1	スタート
2	通系
3	コールプレス5
4	エンド
5	スタート
6	通系
7	コールプレス9
8	エンド
9	スタート
10	通系
11	通系
12	エンド

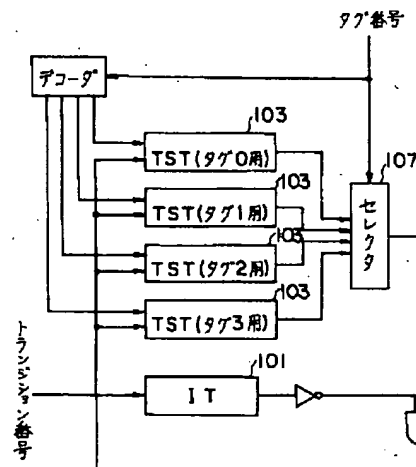
(b) PAT

タグ番号	呼び出し元
1	タグ0.プレス3
2	タグ1.プレス7

(c) タグリスト

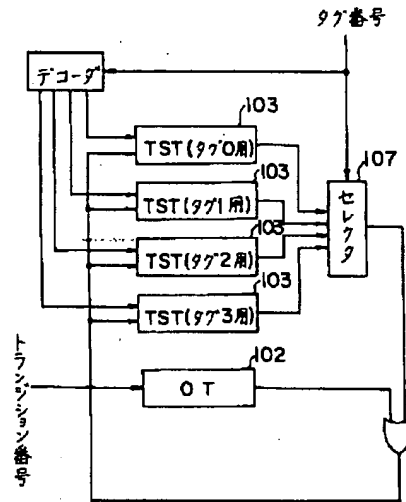
【図12】

従来のタグも導入したTST更新回路



【図13】

従来のタグを導入したTSTトークン注入回路



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-056886

(43)Date of publication of application : 03.03.1995

(51)Int.Cl.

G06F 15/82

G06F 15/16

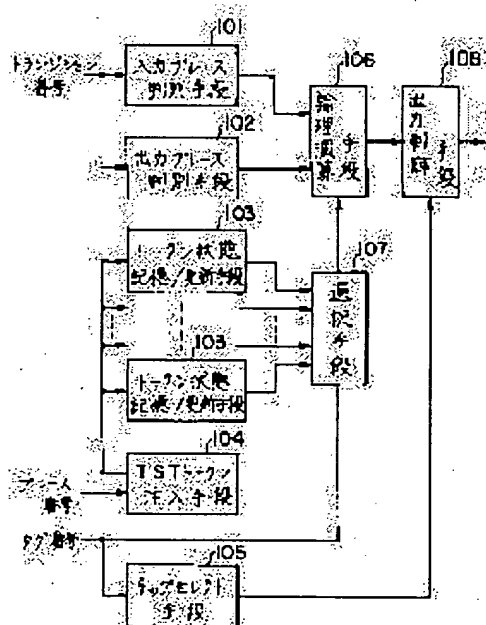
(21)Application number : 05-206292

(71)Applicant : FUJI ELECTRIC CO LTD
FUJI FACOM CORP

(22)Date of filing : 20.08.1993

(72)Inventor : KAWAHARA NORIAKI
ANZAI FUMIHIKO

(54) PARALLEL CONTROL SYSTEM BASED ON PETRI NET



(57)Abstract:

PURPOSE: To make an ignition check even when there are many nesting structure.

CONSTITUTION: An ignition circuit in an execution order controller 503 which controls a state check device 502 which monitors the execution state of a processor in the hierarchical order description is provided with an input place discrimination means 101 which stores the information on the input place to be connected to the transition, output place discrimination means 102 which stores the information of the output place to be connected to the transition, token state storage/updating means 103 storing and updating the state of the token of the place by hierarchy, TST token injection means 104 injecting the token of the place into the token state storage/updating means of each hierarchy, selection means 107 selecting one of plural token state storage/updating means, chip select means 105 selecting the execution order controller, output control means 108 making the output of the execution order controller valid or invalid, and logic

arithmetic means 106 discriminating whether or not the ignition is enabled.

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office